SYSTEM INTEGRATION AND TESTING STANDARDS

Release Date: January 12, 2001

Contract Number: GS-35F-4919H
Order Number: T0000AJ3739

CLIN #007

Prepared for:

U.S. DEPARTMENT OF EDUCATION

OFFICE OF STUDENT FINANCIAL ASSISTANCE (SFA)
400 Maryland Avenue, SW
Washington, DC 20202

Prepared by:

CTGi
Suite 250
10461 White Granite Drive
Oakton, VA 22124

This page left intentionally left blank

FOREWORD

CTG, Incorporated (CTGi) would like to thank the following personnel whose dedication and involvement made the development and completion of this document possible.

Kelly Conboy

Glenn Foster

Candace Jones

Frank Majewski

Annette Mazie

Antony Mosley

Paul Stocks

Sandra Stocks
Director Of Operations,
Washington Metropolitan Area
CTGi

This page left intentionally left blank

EXECUTIVE SUMMARY

This document, prepared for the United States (U.S.) Department of Education, Office of Student Financial Assistance (SFA), provides System Integration and Testing (SI&T) standards for use during the SI&T phase of the SFA Modernization Technology program.

This document, along with those listed below, will be integrated into the U.S. Department of Education, SFA, SI&T Process Handbook, which will then become integrated into the overall U.S. Department of Education, SFA Modernization Technology Handbook. The remaining documents that will comprise the U.S. Department of Education, SFA, SI&T Process Handbook include:

- Test Performance Measurements
- Procedures and Templates for Creating Test Conditions, Test Scenarios, and Testing Data
- Procedures and Templates for Test Execution, Test Evaluation, and Error Correction
- Procedures and Templates for System Configuration Management (CM) and Quality Assurance (QA)
- Procedures For Using Testing Tools

Each of the above listed documents was prepared for delivery, as a separate document.

All SI&T guidelines and procedures are focused on supporting systems and projects used in the development and execution of a comprehensive integration and testing program. To this end, this document contains information on understanding issues related to test planning roles, responsibilities of the organization, documentation requirements, and required testing practices. The standards and procedures used to support this document reflect industry best practices, practices of other federal government agencies, and various governing standards and literature regarding CM and QA procedures during the SI&T process.

This page left intentionally left blank

Table of Contents

Table of Contents (Cont'd)

Table of Contents (Cont'd)

This page left intentionally left blank

LIST OF EXHIBITS

This page left intentionally left blank

LIST OF ACRONYMS

CASE	Computer Aided Software Engineering
CI	Configuration Item
CM	Configuration Management
CMO	Configuration Management Office
COTS	Commercial Off the Shelf
CSCI	Computer Software Configuration Item
CTGi	CTG, Incorporated
DCR	Document Change Request
DID	Data Item Description
FSTR	Final System Test Report
GUI	Graphic User Interface
HWCI	Hardware Configuration Item
IEEE	Institute of Electrical and Electronic Engineers
IDE	Interactive Development Environment
IPT	Integrated Product Team
IRS	Interface Requirements Specification
IVV	Independent Verification and Validation
LOC	Lines of Code
PM	Project Manager
QA	Quality Assurance
QAO	Quality Assurance Office
QC	Quality Control
SD	System Development
SDF	Software Development File
SDS	Software Design Specification
SFA	Office of Student Financial Assistance
SI&T	System Integration and Testing
SPR	System Problem Report
SQT	System Qualification Test
SRD	Software Requirements Document
SRS	Software Requirements Specification

LIST OF ACRONYMS

SSS	System or Subsystem Specifications
STD	System Test Description
STO	System Test Organization
STP	System Test Plan
STR	System Test Report
SU	Software Unit
TM	Test Manager
TPM	Test Performance Measurements
TRR	Test Readiness Review
WBS	Work Breakdown Structure

1. INTRODUCTION

System testing is the principle means to ensure system requirements allocated to software are being satisfied (i.e., testing systems by means of the software). It must be recognized that software testing is a subset of the overall test strategy employed by the customer as a means of ensure the software and/or system meets operational requirements and specifications.

1.1 Background

The United States (U.S.) Department of Education, Office of Student Financial Assistance (SFA), contracted CTG, Incorporated (CTGi) to develop standardized System Integration and Test (SI&T) standards and procedures. These standards and procedures will be used for guidance, planning, and implementation involving current and future U. S. Department of Education, SFA, enterprise information technology systems projects.

1.2 Objective

This document defines standard requirements for procedures designed to support the SI&T process. SI&T standard processes are necessary for the common and systematic integration and testing of all types of information technology systems. The goal of this document is to provide a framework and model leading to development of procedures, activities, and tasks that ensure the U. S. Department of Education, SFA, a consistent approach to measuring and defining system integrity, reliability, efficiency, and functionality.

1.3 Applicability

When the SI&T process is performed by the U. S. Department of Education, SFA, staff and/or contractors, this standard applies, unless specifically excluded, in the program/project plan, contract, etc. This standard is used to create guidelines and procedures for planning, preparation, execution, analysis, and evaluation of all types of U. S. Department of Education, SFA, information technology project integration and testing.

1.4 Assumptions

This document assumes that U. S. Department of Education, SFA, will establish a minimal System Test Organization (STO) similar to the one used in the example in Section 3, Proposed System Test Organization Overview. The STO would be organized to support the SI&T processes during information technology project integration and testing.

The standards in this document represent the minimum Configuration Management (CM) and Quality Assurance (QA) support to be provided to the SI&T process in lieu of established policies of a Configuration Management Office (CMO) and Quality Assurance Office (QAO).

The following assumptions were made:

- SFA has established standards and procedures for CM and QA, as defined in the Modernization Technology Handbook, and that the CM and QA functions encompass the complete system life cycle of SFA individual development efforts.
- A CMO exists within the SFA organization. The CMO is responsible for providing CM support to the Project Manager (PM) for the SI&T process. The standards and procedures described in the CM section of this document and implemented by CM conform to the overall CMO schema.
- A QAO exists within SFA organization. The QAO is responsible for providing QA support to the PM for the SI&T process. The person(s) responsible for QA for a given system development (SD) effort will be assigned on a real-time basis.
- QA personnel will have access to all aspects, formal and informal, of development efforts in order to perform an adequate and accurate assessment.
- QA personnel will have continuous access to developer documentation, status information, CM data, test results, and anomaly data.

1.5 Tailoring

All tasks and activities stated in the SI&T standards may not apply to every project undertaken by U. S. Department of Education, SFA, staff and/or contractor(s). Procedures and templates, derived from SI&T procedures and templates, are a plan and can be tailored for each project, as appropriate. The procedures and templates tailoring process will be performed by the SFA organization and the PM in conjunction with appropriate personnel of the STO and/or Integrated Product Team (IPT). While tailoring of templates and procedures may be necessary to adjust to customer needs, the tailoring must be approached with the clear understanding that the integrity and thoroughness of the SI&T process must remain intact in order to retain and improve the organizational process and standards of the STO and/or IPT. The templates and procedures tailoring effort depends on the type of system being developed (i.e., new or upgrade), the SD methodology, or the software being implemented by the developer.

Procedures and templates tailoring consist of the following:

- Identify requirements that are not applicable.
- Identify additional requirements.
- Provide quantifiable criteria for the requirements (e.g., how often, how many, quality criteria, etc.).

1.6 Terminology

Several terms have specific meaning in this document. These terms are used consistently to emphasize and ensure understanding of the SI&T process, procedures, and requirements. Please refer to the glossary section at the end of this document for definitions.

1.7 Document Organization

This document contains nine narrative sections, a Glossary, a Bibliography, and an appendix. Section 1 provides brief background information and states the guiding objective and applicability for the document. Section 2 provides an overview of the SI&T process. Section 3 summarizes and defines an example STO that could be used during the SFA SI&T process. Section 4 provides Test Performance Measurements procedures. Section 5 provides SI&T Analysis and Planning phase procedures. Section 6 provides Software Unit (SU) Test phase procedures. Section 7 provides Integration Test phase procedures. Section 8 provides Performance Test phase procedures. Section 9 provides System Qualification Test (SQT) phase procedures. Appendix A provides an overview of system testing techniques employed during the SI&T process.

This page left intentionally left blank

2. SYSTEM INTEGRATION AND TESTING PROCESS OVERVIEW

The first part of the SI&T process focuses on identifying the phases required for completion of the process. Each phase of the process has defined entry and exit criteria of accomplishments, documents, and/or event(s). The five phases of the SI&T process are shown in Figure 2.1.

2.1 System Integration and Testing Process Phases

Figure 2.1 depicts the phases of the SI&T process.

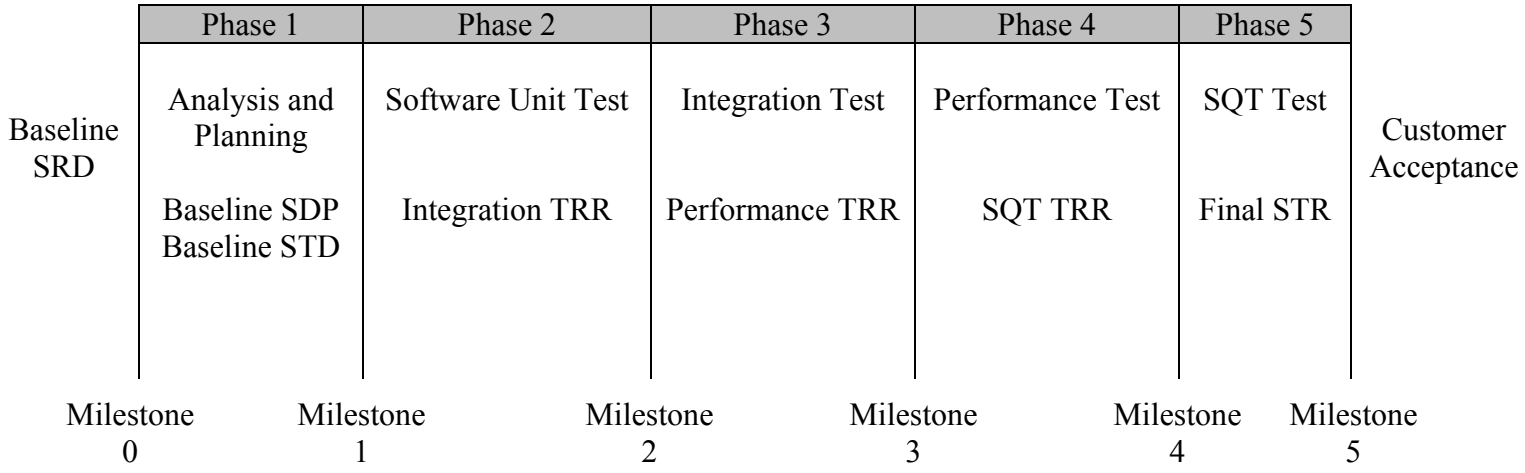

Figure 2.1 System Integration And Testing Process Phases and Milestones

Criteria required to accomplish Milestone 0 and begin the SI&T process are the completion and approval by the SFA of a baseline Software Requirements Document (SRD) or an equivalent document. The many efforts, actions, and organizations needed to accomplish this milestone will not be referenced or documented here as they are outside the scope of the SI&T process. The baseline SRD is a formal document derived from the Software Requirements Specification (SRS), Interface Requirements Specification (IRS), System or Subsystem Specifications (SSS), etc. that set the requirements, specifications, and standards for a system. The SRD contains functional specifications and requirements, performance specifications and requirements, interface specifications and requirements, design specifications and requirements, and development requirements and standards.

2.1.1 Analysis and Planning Phase

During the Analysis and Planning Phase of the SI&T process, the entry criteria is the completion of a baseline SRD and the exit criteria is the completion of a baseline System Test Plan (STP) and a baseline System Test Description (STD). The SU Test phase may begin upon completion and baseline of the STP.

2.1.2 Software Unit Test Phase

The SU Test phase of the SI&T process, the entry criteria are the approval and baseline of the STP. During this phase, see Figure 2.2, unit, unit integration, and Computer Software Configuration Item (CSCI) qualification tests are completed. The objective of the SU Test phase is to verify satisfaction of CSCI performance requirements, as documented in the SRD. STD creation continues during this phase, if not completed before this phase began. The exit criterion of phase two is the completion of the Test Readiness Review (TRR) for Integration Test with authorization to proceed to the Integration Test phase.

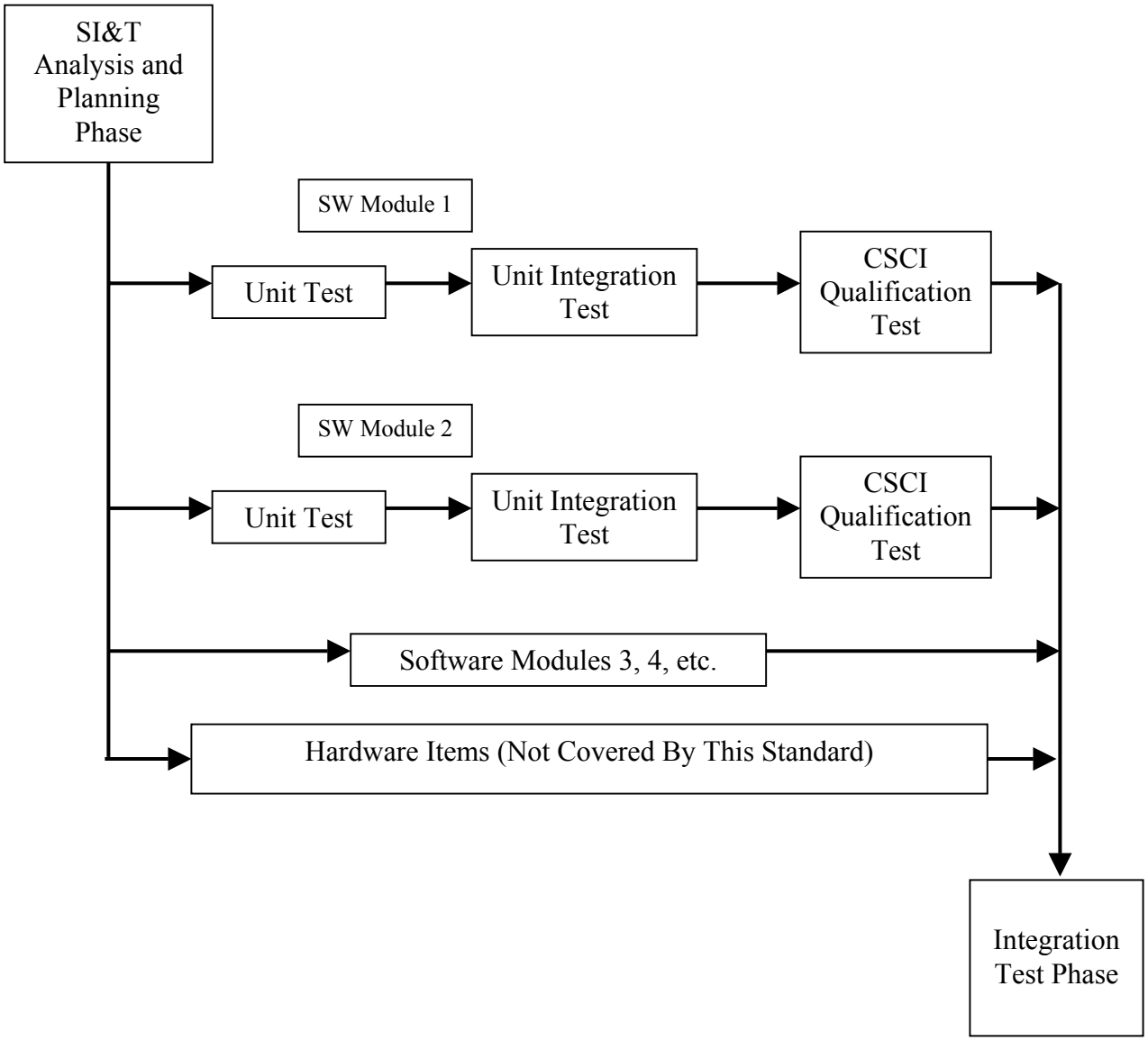

Figure 2.2 Software Unit Test Phase Overview

2.1.3 Integration Test Phase

Entry criteria for the Integration Test phase of the SI&T process is authorization to proceed by the TRR at the completion of the SU Test phase and completion and baseline of the STD and the baseline STP. During the Integration Test phase, the qualified CSCI modules are integrated and regression testing is completed, when necessary, in accordance with the STP and the applicable STD. The objective is to validate that Hardware Configuration Item (HWCI) and CSCI components can be individually interfaced. The exit criterion of Integration Test phase is completion of the TRR for Performance Test with authorization to proceed to the Performance Test phase.

2.1.4 Performance Test Phase

Entry criteria for the Performance Test phase of the SI&T process is authorization to proceed by the TRR at completion of the Integration Test phase. During the Performance Test phase, the integrated software is incorporated into the environment of the production system. The objective of this phase is to test system processing and response time (i.e., volume and stress test) during peak usage. The exit criterion for the Integration Test phase is completion of the TRR for SQT, with authorization to proceed to the SQT phase.

2.1.5 System Qualification Test Phase

Entry criteria for the SQT phase of the SI&T process is authorization to proceed by the TRR at the completion of the Performance Test phase. During the SQT phase, all testing is completed in a production environment. The objective of this phase to demonstrate system functionality, ease of operator operation, and compliance with all requirements set forth in all design specifications and documents. The exit criterion for the SQT phase is completion, and approval by the PM, of the Final System Test Report (FSTR) and customer acceptance of the tested system.

2.1.6 Post Testing Phase

After PM approval of the FSTR and formal acceptance of the tested system by the customer or customer's representative, the SI&T process is completed. Tests that follow the SQT are typically conducted and performed by agencies and/or organizations external to the STO. Processes for these tests are not discussed in this document.

2.1.7 External Interface Testing

If requirements state that external interface testing is necessary before a system can be released from the SI&T process, the external interface testing can be accomplished during the Performance Test phase, the SQT phase, or both. The testing of a system's external interface(s) can be implied from the "volume and stress testing" performed during the Performance Test phase. External interface(s) testing can also be implied during "…tests that demonstrate system

functionality, or ease of operator operation, and compliance with all requirements set forth in all design specifications and documents" performed during the SQT phase. External interface testing of a system in these phases can be accomplished by the use of a simulator/emulator or other device(s) to act as the receiver/transmitter on the other side of the interface. If testing during these two testing phases is rigorous and external interface specifications are defined for all circumstances, there is less chance of a problem(s) later in the system life cycle during communication with an external system(s).

The decision of when, where, and how to accomplish external interface testing during the SI&T process can vary. Testing of the external interface(s) should be tailored to each system after discussions, planning, and coordination between the PM, test manager, the customer, and other SI&T personnel.

2.2 System Integration and Testing Process Overview

An overview of the SI&T process that shows the relationship of documents, test phases, and the CM and QA support is provided in Figure 2.3.

The process is constructed for only one formal input document, the SRD or an equivalent document, to be used as the basis for testing activities and procedures. Management support and guidance for CM and QA activities and procedures come from the CMO and QAO, as depicted in Figure 3.1, respectively. The last activity of the SI&T process is the approval of the FSTR. The PM and the customer must sign the FSTR before the final phase of the SI&T process is concluded and formal responsibility of the information technology system is removed from the STO.

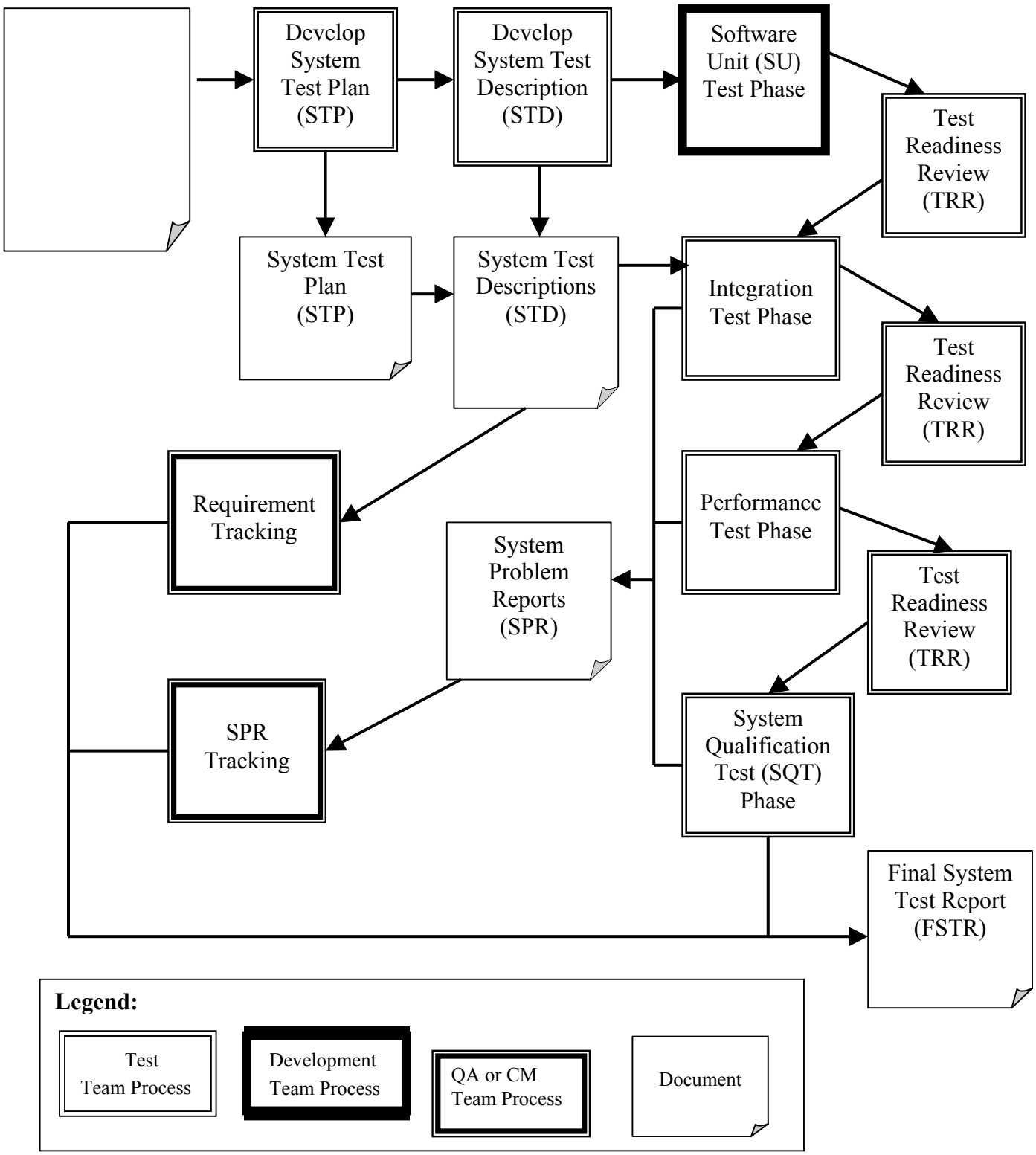

Figure 2.3 System Integration And Testing Process Overview

This page left intentionally left blank

3. PROPOSED SYSTEM TEST ORGANIZATION OVERVIEW

Adapting a system testing discipline for the production of software-intensive systems requires more than just an understanding of the technical issues. To develop a testing capability, begin by establishing system test standards and policies that define system testing. An additional concern is coordinating system test activities to support the needs and priorities of the process, satisfy the customer, and achieve the overall objectives of testing. To that end, responsibility for testing activities must be established.

The following paragraphs discuss a proposed STO structure. This proposed organization is presented in Figure 3.1.

Error! Not a valid link.

Figure 3.1 Proposed System Test Organization

The STO structure outlined in this section is based on the implementation of developed standardized procedures for conducting the SI&T process. There processes must be established and documented that provide members of the STO the information required to interpret and execute the SI&T process standards. Prescribed practices and procedures should encompass administrative, analysis, planning, review, development, integration, testing, documentation, CM, and QA. A well-organized STO assists in ensuring that the following requirements are met:

- Test planning, design, integration, execution, and evaluation responsibilities are explicitly assigned.
- Test objectives and phases of tests to be conducted are documented in an STP.
- Test case(s) and test case procedure(s) have been developed for all formal-testing phases and documented in an STD.
- Test resources and artifacts are under CM control.
- Testing activities and documentation are reviewed and evaluated by QA.

The development of test policies and standard test processes are performed or coordinated by designated personnel. Coordination of the testing process ensures cycle activities are performed, documented, reviewed, and approved in accordance with approved test policies and standards throughout the testing life.

3.1 Executive Manager

The Executive Manager is generally an organizational or enterprise manager responsible for multiple projects. This manager defines high-level performance and business objectives and ensures that individual projects support the overall organizational strategy.

3.2 Project Manager

The PM oversees all personnel, technical, quality, cost, and schedule aspects of the SI&T process. In this context, the PM has overall responsibility for the completeness of the development and testing efforts. The PM is the principle point of contact (POC) for the entire STO, including CM, QA, testing, and SD groups. The PM provides direction and management controls to ensure the success of project activities and resolution of issues that may inhibit SI&T process efforts.

Within the STO, depending on the size and scope of the project, the PM meets with various personnel from supporting organizations (e.g., the Executive Manager, CMO, and QAO) to plan, prioritize, and coordinate testing tasks and procedures within each organization. In this way, standards and procedures developed for the project life cycle are incorporated into the SI&T process. The PM then has the flexibility to assign tasks to personal with appropriate expertise.

3.3 Quality Assurance Group

QA is a process that evaluates the form, structure, and/or compliance of a system in relation to applicable standards. QA provides a standard that ensure all Quality Control (QC) requirements are implemented during the SI&T process. The QA group provides evaluation disciplines to the SI&T process.

The QAO is the officiator of the QA system life cycle process. The QAO develops standards and procedures used to implement QA functionality throughout the SI&T process. Although the QAO is responsible for reviewing and auditing the entire system project, this sub-section only discusses QA as it relates to SI&T and the form, structure, and/or compliance with the SI&T processes.

The QA group reports directly to the PM. The QA group provides the PM with assurance that all QC requirements for SI&T processes are met by conducting reviews and/or audits of testing activities. The QA group generates both formal and informal reports from the results of the reviews and/or audits. Throughout the SI&T process, QA responsibilities include providing the following information to be used in the Test Performance Measurements (TPM) process:

- Evaluation of the system/test corrective action process.
- Evaluation of the CM process.
- Evaluation of all system/test documentation.
- Evaluation of all guidelines, test cases, and test case procedures.
- Creation of measurement information.

3.4 Configuration Management Group

CM controls the integrity of the project being developed and tested. CM is the vehicle by which changes/modifications made to documentation, software, and environments during the life cycle of the SI&T process are managed and controlled. The CM group provides the disciplines that apply technical and administrative direction and control to the SI&T process.

The CMO is the officiator of the CM system life cycle process. The CMO develops standards and procedures used to implement CM functionality throughout the SI&T process.

The CM group reports directly to the PM and executes baseline and version control, identification, and change control standards and procedures throughout the SI&T process. The CM group generates both formal and informal reports based on progress of the testing life cycle. Throughout the SI&T process, CM responsibilities include providing the following information, to be used in the TPM process:

- Ensure all formal software, hardware, and test environment configurations, as specified in the SRD and STP, are placed under CM control.
- Ensure all formal documentation is placed under CM control.
- Ensure formal release procedures for CM approved software, documentation, and test environment versions are established.
- Ensure unauthorized changes to the controlled software, documentation, and test environment are prevented, and ensure incorporation of all approved modifications/changes.
- Track and report system problems.
- Create measurement information.

3.5 System Development Group

The SD group interacts directly with the Test group, but remains a separate entity within the STO. This separation ensures the objectivity of the test group in evaluation of system requirements and quality.

Within the realm of testing, the SD group is responsible for analysis, design, testing, implementation, quality, and documentation of SU Test phase testing. The SD group reports schedule, quality, technical performance, and documentation of the SU Test phase effort to the PM. In addition, the SD group ensures that policies and procedures established for the Software Development Files (SDF) are followed and provides comprehensive SU Test phase coverage, which contributes to the overall success of the testing process.

During Integration Test, Performance Test, and SQT phases, the SD group is responsible for the repair of all software problems reported and cooperates with the testing group to ensure the reported problems are solved and tested in an expeditious manner.

The SD group provides requested measurement information regarding all efforts, activities and responsibilities during the SI&T process to the PM.

3.6 Test Manager

The Test Manager (TM) manages and guides all integration and testing functions of the SI&T process. The TM is responsible for specification of test standards, allocation of resources, test scheduling, and management of test engineers. Additional TM responsibilities include the following:

- Develop software test process standards in a concise and usable form and a process to communicate these standards to other groups of the STO.
- Create software test methodologies.
- Develop specified standard tools and technologies to support SI&T processes.
- Schedule time and resources for both hardware and software testing.
- Resolve issues that may obstruct or inhibit the testing schedule or effort.
- Collect measurement information requested by the PM.

3.6.1 Test Engineer

The test engineer(s) is responsible for testing the system during the SI&T process. The test engineer(s) is responsible for development and creation of the STD. It is the responsibility of the test engineer(s) to ensure STP and STD tasks are implemented during the SI&T process. The test engineer(s) performs Integration Test phase and Performance Test phase testing. During the SQT phase the test engineer(s) may or may not perform the testing. If the test engineer(s) does not perform testing during the SQT phase, the test engineer(s) shall act as testing witness(es). As test case procedures are executed the test engineer(s) communicates testing status to the TM.

The test engineer(s) authors the STR at the end of the Integration Test, Performance Test, and SQT phases of the SI&T process. The test engineer(s) documents problems encountered during testing using the System Problem Report (SPR) process. The test engineer(s) is responsible for re-testing SPRs after they are returned from the SD group. After re-testing, the test engineer(s) either recommends closure of the SPR and forwards the recommendation to the SPR formal review process panel or returns the SPR to the SD for further investigation and repair(s).

This page left intentionally left blank

4. TEST PERFORMANCE MEASUREMENTS

TPM implementation provides objective information to address project issues by integrating TPM with existing risk management and performance management disciplines.

Measurement is a key element of successful management in every well-established engineering discipline. TPM present an approach for tailoring and implementing an effective measurement process for software-intensive projects. The objective is to provide PMs with the system information required to make informed decisions that impact project cost, schedule, and technical objectives.

TPM describes system measurement as a systematic, flexible process that is an integral part of the overall project management structure. Project issues drive the TPM process. While detailed treatments of risk management and financial performance management are beyond the scope of this document some understanding of these topics is necessary to gain the full benefit of TPM. Figure 4.1 presents an overview of the quantitative TPM process.

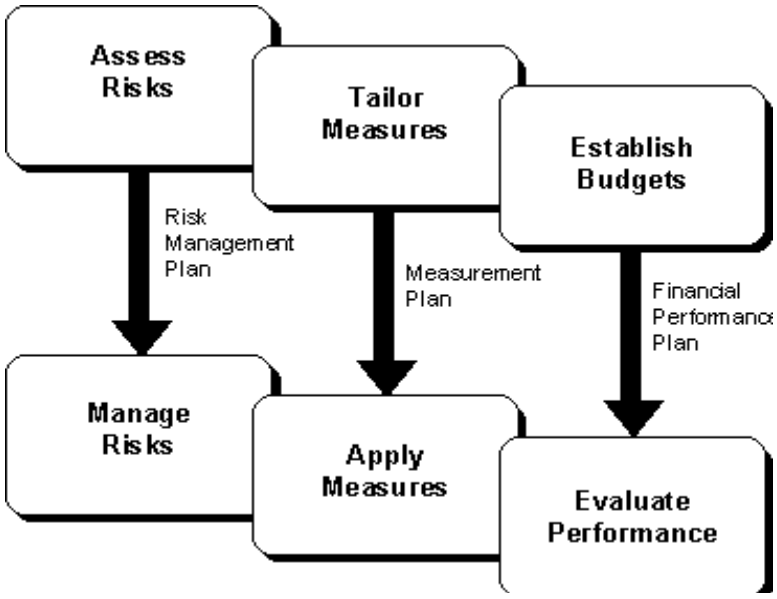

Figure 4.1 Quantitative Test Performance Measurements Process Overview

The TPM process is adaptive to meet the specific information needs and characteristics of each individual project. The process is based on a proven set of system measurement principles listed in Section 4.3,whch are derived from actual experience on government and industry projects. These principles represent measurement "best practices" that make the TPM process an effective management tool, not just another project management "requirement."

4.1 Test Performance Measurements During System Integration and Testing Process

During the SI&T process, the project focus is on getting the system ready for delivery to the customer. This usually means that the focus is on evaluating system quality. The SI&T process is often the shortest and most intense activity in a system life cycle. Consequently, the TPM process must focus on providing rapid data collection, analysis, and feedback to project management so effective decisions can be made in a timely manner. The procedures, activities, and tasks required to ensure a consistent approach for measuring and defining system integrity, reliability, efficiency, and functionality should be utilized. On many projects, this results in increased analysis of documentation and various reports. A weekly reporting interval for SPRs often is used during the SI&T process. In some cases, daily reporting of test progress and SPRs is provided.

The determination of the reporting interval depends on many factors, but there is usually an increase in measurement activity during the SI&T process. Effective TPM give an accurate and comprehensive assessment of testing activities, minimize and standardize the burden of data collection, and is accepted and used to improve the SI&T process performance.

Each project is described by differing management and technical characteristics and by a specific set of system issues and objectives. To address the unique measurement requirements of each project, refer to U. S. Department of Education, SFA, document "Test Performance Measurements." This document explains actions, activities, and procedures necessary for tailoring and applying a defined TPM process to meet specific project information.

4.2 Test Performance Measurements Implementation Roles

The TPM process is an integral part of the system and/or software testing process. Many members of the STO play important roles. Appropriate resources must be allocated for the TPM process to work effectively.

The most important roles in the TPM process are listed below:

- **Executive Manager.** The Executive Manager is generally an organizational or enterprise manager responsible for multiple projects. This manager defines higher-level performance and business objectives and ensures that individual projects support the overall organizational strategy. TPM helps the Executive Manager determine the status of individual projects and make decisions that apply across the organization.
- **Project Manager.** The PM is responsible for identifying issues, reviewing analysis results, and acting on measurement information. In the optimal case, the SD and Test groups each have a PM who uses TPM information to make decisions for their respective group and to communicate objectively between groups.

- **System Test Organization**. The STO is responsible for day-to-day development and testing of a system or software application. The SD, Test, QA, and CM groups within the STO can be comprised of government and industry personnel or organizations, and may be defined within an IPT structure. STO groups are responsible for collection of measurement data on a periodic basis and all groups use the measurement results to guide SI&T activities.
- **Measurement Analysis Personnel.** This role can be assigned to an individual or a team. Responsibilities include developing the project TPM plan, if necessary, collecting and analyzing measurement data, and reporting results throughout the STO. Each organization within the STO that makes critical system decisions should have an independent measurement analysis capability.

Ensure that all participants in the TPM process understand and commit to their responsibilities. This ensures that accurate information is available to support effective communications and informed decision making.

4.3 Test Performance Measurement Principles

To accomplish the unique measurement requirements needed to implement the gathering of specific project information, nine principles define the TPM system measurement process and describe the characteristics for an effective process . These principles are:

- Project issues and objectives drive measurement requirements.
- Software developer processes defines how the software is actually measured.
- Collect and analyze data at a level of detail sufficient to identify and isolate system problems.
- Implement an independent analysis capability.
- Use a systematic analysis process to trace measures to decisions.
- Interpret measurement results in the context of other project information.
- Integrate system measurements into the project management process throughout the system life cycle.
- Focus initially on project-level analysis.

This page left intentionally left blank

5. ANALYSIS AND PLANNING, PHASE ONE

5.1 Phase Purpose

The purpose of the Analysis and Planning phase of the SI&T process is to determine and document standardized system testing procedures and conditions for use during system testing. Security requirements, if necessary, shall be considered during the creation of all documentation during this phase. The Analysis and Planning phase begins when the STO receives a baseline SRD or equivalent document. The objective of this phase is to create a baseline STP and a baseline STD.

The STP shall describe procedures for qualification testing of CSCI and HWCI systems. The STP describes the system test environment, identifies tests to be performed, testing procedure (i.e., manual or automated test tool), and provides schedules for test activities. The STP enables the customer and STO groups to assess the adequacy of the testing phases of the SI&T process.

The STD describes test preparations, test cases, test case procedures, and traceability of testing requirements to the SRD for the qualification testing of CSCI and HWCI systems. The STD describes the sequence of testing, the method of testing, and the criteria regarding the Pass/Fail status of a test case. In most circumstances, only one STD is created per project; however, if the system is complex, more than one STD maybe created during this phase. The STD enables the customer and STO groups to assess the adequacy of qualification testing performed during the SI&T process.

Another purpose of the Analysis and Planning phase is to create or update a Software Development File (SDF), which is used only during the SU Test phase of the SI&T process. SDF creation or upgrade(s) is the responsibility of the SD group or developer. New SDF(s) creation or upgrade(s) of an existing SDF(s) will be completed in accordance with STP requirements and other standards, requirements, and procedures, as required. When it is necessary to create a new SDF, for creation of a new system or an upgrade to a system, the format, type, and quantity of data necessary to satisfy testing and documentation requirements must be determined before completion of this phase.

5.2 Entry and Exit Criteria

Entry criteria must be satisfied and all exit criteria must be completed for successful completion of this phase of the SI&T process.

5.2.1 Entry Criteria

Entry criteria for the Analysis and Planning phase of the SI&T process is receipt of a baseline SRD or equivalent document by the STO.

5.2.2 Exit Criteria

For new SDF creation or an SDF upgrade(s), the format, type, and quantity of data necessary to satisfy testing and documentation requirements must be defined as part of the exit criteria for this phase.

Exit criteria for the Analysis and Planning phase of the SI&T process is:

a. Completion, baseline approval, and submission of the project STP to CM.

b. Completion, baseline approval, and submission of the project STD(s)to CM.

5.3 Test Group Activities

The PM appoints a person from the Test group to lead and coordinate the process of creating a baseline STP and a baseline STD. For templates for the creation of these documents, refer to the Appendices A and B of the U.S. Department of Education, SFA, document "Procedures and Templates For Test Creation."

For a detailed description of Test group activities during the Analysis and Planning phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Creation."

5.4 System Development Group Activities

The SD group initiates and supports required activities in coordination with other groups in the STO during the creation of a baseline STP and baseline STD(s).

5.4.1 Software Development File Creation and Upgrades

The SD group or developer personnel are responsible for new SDF creation or SDF(s) upgrade(s). All new or upgraded SDFs will be completed in accordance with STP requirements and other standards, requirements, and procedures, as required. SDFs may be generated, maintained, and controlled by manual or automated means. The SD group or developer shall document and implement procedures to establish and maintain the SDFs for each of the three parts of the SU Test phase.

When a new SDF is created or an SDF is upgraded, it shall include (directly or by reference), at minimum, the following information:

- Design considerations and constraints.
- Design documentation and data.

- Scheduling and status information.
- Test requirements and responsibilities.
- Test case, test case procedures, and results.

Additional information and/or data required to support additional unique or TPM information requirements will be provided with the above minimum information requirements.

5.5 Configuration Management Group Activities

The CM group places both the baseline STP and baseline STD under version control and assign version identification to the documents.

For a detailed description of CM group activities during the Analysis and Planning phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 3.

5.6 Quality Assurance Group Activities

The QA group monitors activities of the Test and SD groups or developers to verify that system requirements are appropriately documented in the baseline STP and baseline STD.

The QA group monitors for compliance to the STP and other standards, requirements, and procedures for new SDF creation or an SDF upgrade(s), if directed by the PM.

For a detailed description of QA group activities during the Analysis and Planning phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 6.

This page left intentionally left blank

6. SOFTWARE UNIT TEST, PHASE TWO

6.1 Phase Purpose

The SU Test phase verifies satisfaction of CSCI performance requirements, as documented in the SRD. The purpose of the SU Test phase is to identify and correct internal logic errors contained in unit, unit integrated, or CSCI-qualified software modules. For an overview of the SU Test phase, refer to Figure 2.2. For an explanation of this type of "white box," testing refer to Appendix A, Testing Techniques.

The SU Test covers the areas of path testing and boundary condition testing. Path testing is defined as testing that executes every logic branch and line of code to find logic errors in control structures, dead code, errors in loop boundaries, and errors in loop initializations. This includes every state and every mode. Boundary condition testing is defined as testing performed to find errors in input and output parameter tolerances and verify that the program limits are correctly stated and implemented.

The unit integration test function verifies correct operation and function as the units are incrementally integrated to form larger and more complex software modules (i.e., CSCI modules). Integration activities continue until all required units are integrated into functioning CSCI modules, as defined in the STP.

The CSCI qualification test verifies satisfaction of CSCI requirements, as documented in the SRD. CSCI qualification testing covers path testing and boundary condition testing.

Actual hardware or the equivalent that will become the HWCI is used for testing during this phase, whenever and wherever possible.

6.2 Entry and Exit Criteria

Entry criteria must be satisfied and all exit criteria must be completed for successful completion of the SU Test phase of the SI&T process.

6.2.1 Entry Criteria

Entry criteria for the SU Test phase are completion, baseline approval, and submission of the project STP to CM.

6.2.2 Exit Criteria

Criteria for exit from the SU Test phase of the SI&T process are:

a. Completion of qualification testing of all required system CSCI modules.
b. Submission to CM of the source code of all CSCI qualified modules.
c. Submission to CM of all completed testing SDFs.
d. Submission to CM of all necessary HWCI information.
e. Authorization of the Integration Test TRR to proceed to the Integration Test phase.

6.3 Test Group Activities

No formal, "black box," testing will be undertaken by the Test group during the SU Test phase of the SI&T process.

During the SU Test phase, the Test group is responsible for activities leading to the creation of test cases, test case procedures and automated test scripts for use in the later formal "black box" testing phases. The Test group coordinates activities and needed support for this task with the SD group.

Additional responsibilities during the SU Test phase include the creation, preparation, and configuration of the infrastructure necessary when using an automated testing tool, if a testing tool was specified for use in the project STP. For activities, check-off lists, and suggested tables for completion of these tasks refer to the U.S. Department of Education, SFA, document "Procedures For Using Testing Tool" and associated appendices.

If activities, problems occur during the SU Test phase that require change(s) and/or modification(s) to a governing standard, plan, template or guideline, CM must be notified prior to implementation of the change(s) and/or modification(s). Test group personnel will provide CM with a description of the problem, the cause of the problem, and the documentation change(s) required. No change(s) and/or modification(s) to a governing standard, plan, template, guideline, or previously created documentation under CM version control shall be undertaken until notification of approval for the change(s) is received from the CM group.

Test group personnel will participate in the Integration Test TRR that convenes at the end of the SU Test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of Test group activities during the SU Test phase of the SI&T process, refer to U.S. Department of Education, SFA, documents "Procedures and Templates For Test Execution," Section 3, and "Procedures For Using Testing Tool," Section 2.

6.4 System Development Group Activities

The SD group or developer will act as the lead group during the SU Test phase.

The SD group aids with needed activities and provides support to the Test group during the design and creation of formal test cases, test case procedures, and automated test scripts.

The SD group or developer creates and tests all new or modified CSCI software modules needed to complete this task. All software coding is governed by the SFA and/or project software coding standards and guidelines. System(s) database creation follows SFA database design guidelines. Security requirements, if necessary, shall be considered during the creation of all coding, presentations, and documentation during this phase.

If problems occur during the coding process that require the change(s) and/or modification(s) to a governing standard, plan, template or guideline, CM must be notified prior to implementation of the required change(s) and/or modification(s). SD or developer personnel will provide CM with a description of the problem, the cause of the problem, and the documentation change(s) required. No change(s) and/or modification(s) to a governing standard, plan, template, guideline, or previously created documentation under CM version control shall be undertaken until notification of approval for the change(s) is received from the CM group.

All problems encountered and their solutions during the SU Test phase will be documented in the SDF being used by the unit or module at that time. No SPR(s) will be initiated during this testing phase.

The SD group will complete the creation, modification(s), or upgrade(s) for all SDFs, as needed, based on the format and requirements, see Section 5.4.1 for details, developed during the Analysis and Planning phase. A separate SDF will be initiated and completed for each unit test, unit integration test, and CSCI-qualified tested software module to support all testing and testing results. All SDFs initiated and completed during the SU Test phase will be provided to the CM group as test artifacts.

SD group personnel will participate in the Integration Test TRR that convenes at the end of the SU Test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of SD group activities during the SU Test phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Section 3.

6.5 Configuration Management Group Activities

CM approve and track, by version control, all changes and/or modifications to all previously received governing standards, plans, templates, guidelines, documents, and documentation that occur during the SU Test phase. All version control changes and associated information will be reported to the PM in a timely manner and will become a part of the record of the Integrated TRR.

The CM group places the baseline, CSCI-qualified software modules source code, all completed testing SDFs, and HWCI under version control and assign version identification to the test artifacts, documents, and documentation.

CM group personnel will participate in the Integration Test TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of CM group activities during the SU Test phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 4.

6.6 Quality Assurance Group Activities

The QA group monitors the activities of the Test, CM, and SD groups or developers to verify compliance with all SU Test phase standards, plans, templates, and guidelines. Formal or informal inspections or receipt of standard documentation will verify activity compliance.

If directed by the PM, the QA group formally or informally monitors testing activities relating to the CSCI modules during the CSCI qualification test portion of the SU Test phase for compliance to the STP and other standards, requirements, and procedures.

QA group personnel will participate in the Integration Test TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of QA group activities during the SU Test phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 7.

This page left intentionally left blank

7. INTEGRATION TEST, PHASE THREE

7.1 Phase Purpose

The purpose of the Integration Test phase of the SI&T process is to ensures the CSCIs link and operate together. The focus is on the effectiveness of functional interactions and compatibility at the interfaces of the CSCI modules. To ensure integration of functionality derived from the design of the software, testing activities will involve the formal testing of various qualified functionally related CSCIs.

The Integration Test phase can either be an integration of various qualified CSCI modules leading to the creation of a new system or it can be for one or more qualified CSCI modules being integrated into a production system for an upgrade or modification. If a CSCI module(s) is used to upgrade or modify a production system, a regression test must be performed.

If the Integration Test phase is for creation of a new system, only approved CSCIs required to satisfy all functions stated in the SRD are integrated to create the system. The test team tests and verifies the compatibility of each new CSCI with previously integrated CSCIs. This process continues until all required CSCIs are integrated into the system.

When a previously released system, which is being modified or upgraded by the addition of CSCI(s), both the CSCI and the previously released system will undergo regression testing.

The purpose of regression testing is the selective re-testing of the modified or upgraded system to detect faults introduced or caused by the modification(s) or upgrade(s). Regression testing verifies that the modified or upgraded system continues to operate and function to original requirements and specifications.

Security requirements, if necessary, shall be considered during the performance of all activities during this phase.

7.2 Entry and Exit Criteria

Entry criteria must be satisfied and all exit criteria must be completed for successful completion of the Integration Test phase of the SI&T process.

7.2.1 Entry Criteria

Entry criteria for the Integration Test phase are the completion, baseline approval, and submission of the project STP and STD to CM and authorization from the Integration Test TRR to proceed.

7.2.2 Exit Criteria From Computer Software Configuration Item Integration Testing

Criteria for exit from the CSCI integration-testing portion of the Integration Test phase and entry into the regression-testing portion of the Integration Test phase are:

a. Review and approval of the CSCI integration-testing portion of the STR.

b. Submission of SPR forms for all detected problems.

c. Successful completion of CSCI integration testing.

Criteria for exit from the CSCI integration-testing portion of the Integration Test phase and entry into the Performance Test phase are:

a. Review and approval of an Integration Test STR.

b. Submission of SPR forms for all detected problems.

c. Successful completion of CSCI integration testing.

d. Authorization of the Performance Test TRR to proceed to the Performance Test phase.

e. System submitted to CM.

7.2.3 Exit Criteria From Regression Testing

Criteria exit for exit from the regression-testing portion of the Integration Test phase and entry into the Performance Test phase are:

a. Review and approval of an Integration Test STR.

b. Submission of SPR forms for all detected problems.

c. Successful completion of regression testing.

d. Authorization of the Performance Test TRR to proceed to the Performance Test phase.

e. System submitted to CM

7.3 Test Group Activities

The Test group acts as the lead group and is responsible for system testing during the Integration Test phase.

The STD for the regression-testing portion of the Integration Test phase will be developed from previously executed test cases covering mission-critical functions identified in the SRD or equivalent document. To ensure overall operational effectiveness of the production system specific mission-critical functions will be chosen for re-testing. Modifications to previous test

cases and test case procedures will be at the discretion of the STO. All modifications to previous test cases and test case procedures will be documented and submitted to CM and QA.

All tests conducted during the Integration Test phase will be of a formal "black box" type. The Test group, following the guidelines of the STP and STD, will execute all test case procedures in accordance with the associated test case and record all testing results, as they are observed.

All problems encountered during the Integration Test phase of the SI&T process will be documented and reported using an SPR form. After initialization of an SPR, it shall be submitted for review and analysis during the SPR Review process. Refer to U.S. Department of Education, SFA, document "Procedures For Using Testing Tool," Section 2.7, for an explanation of the SPR review process. For the template and instructions for completion of the SPR form, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix D.

If the SD group has resolved a reported problem that was caused by the software, the test case procedure or test script that caused the problem will be re-tested using the original test case. If the repair to the software is tested and found not to satisfy the Pass/Fail criteria established by the STD for the function being repaired or tested, the Test group will not recommend the SPR for closure. Upon completion of the re-test, the SPR will be returned to the SPR Review process with the recommendation that the SPR be returned to the SD group for further investigation and repair. The process of repair and test will be repeated until the implemented repair allows the repaired function to satisfy the STD-established Pass/Fail criteria. When the implemented repair allows the repaired function to satisfy the STD-established Pass/Fail criteria, the Test group will recommend to the SPR Review process that the SPR status be changed to closed.

Upon completion of the Integration Test phase, an STR will be created by the Test group, with the cooperation of the SD and QA groups, and submitted to the Performance TRR and the CM group. For the template and instructions for completion of the STR form, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix C.

If problems occur during the Integration Test phase activities that require a change(s) and/or modification(s) to a governing standard, plan, template, or guideline, CM must be notified prior to implementation of the required change(s) and/or modification(s). Test group personnel will provide CM with a description of the problem, the cause of the problem, and the documentation change(s) required. No change(s) and/or modification(s) to a governing standard, plan, template, guideline, or previously created documentation under CM version control shall be undertaken until notification of approval for the change(s) is received from the CM group.

Test group personnel will participate in the Performance Test TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of Test group activities during the Integration Test phase of the SI&T process, refer to U.S. Department of Education, SFA, documents "Procedures and Templates For Test Execution," Section 5, and "Procedures For Using Testing Tool," Section 2.

7.4 System Development Group Activities

During the Integration Test phase, the SD group will coordinate its activities with the Test group to provide support and be prepared to expedite the repair of all SPRs that require change(s), modification(s), and/or upgrade(s) to the software of CSCI modules or the production system that is under test.

If problems occur during the Integration Test phase activities that require a change(s) and/or modification(s) to a governing standard, plan, template, or guideline, CM must be notified prior to implementation of the needed change(s) and/or modification(s). SD group or developer personnel will provide CM with a description of the problem, the cause of the problem, and the documentation change(s) required. No change(s) and/or modification(s) to a governing standard, plan, template, guideline, or previously created documentation under CM version control shall be undertaken until notification of approval for the change(s) is received from the CM group.

At the end of the Integration Test phase, the SD group will coordinate and cooperate with the Test group in the creation of the STR.

SD group personnel will participate in the Performance Test TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

7.5 Configuration Management Group Activities

The CM group will place the baseline system source code and HWCI under version control and assign version identification to test artifacts, documents, and documentation, as received or at the completion of the Integration Test phase. All "closed" SPRs will be submitted to CM and will be classified as documents for CM accounting purposes.

CM will approve and track, by version control, all changes and/or modifications to all previously received governing standards, plans, templates, guidelines, documents, and documentation that occurred during the Integration Test phase. All version control changes and associated information will be reported to the PM in a timely manner and will become a part of the record of the Performance TRR.

CM group personnel will participate in the Performance Test TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of CM group activities during the Integration Test phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 4.

7.6 Quality Assurance Group Activities

The QA group monitors the activities of the Test, CM, and SD groups or developers to verify compliance with all Integration Test phase standards, plans, templates, and guidelines. Formal or informal inspections or receipt of standard documentation will verify activity compliance.

The QA group will certify the proper system configuration and environment as required in the STP.

At the end of the Integration Test phase, the QA group will coordinate and cooperate with the Test group in the creation of the STR.

QA group personnel will participate in the Performance Test TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of QA group activities during the Integration Test phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 7.

This page left intentionally left blank

8. PERFORMANCE TEST, PHASE FOUR

8.1 Phase Purpose

The purpose of the Performance Test phase of the SI&T process is to validate the compatibility of all hardware and software components in accordance with the guidelines contained in the STP and STD. The Performance Test phase will be conducted in an environment simulating real world usage of the system. Performance testing is an extension of the Integration Test phase and a volume/stress test of the system. During the Performance Test phase, system processing and response time will be documented under peak transaction loading. The testing will validate the operation(s) of all interface and data exchange parameters.

Test cases and test case procedures will provide for processing predicted user volume and unpredicted or excessive user volume. The Performance Test will also test automated processes routinely initiated by the system or automated processes initiated by system users. Automated processes will be tested for normal and excessive volume. Response times for normal and excessive usage will be documented to ensure compliance with SRD specifications.

Security requirements, if necessary, shall be considered during the performance of all activities during this phase.

8.2 Entry and Exit Criteria

Entry criteria must be satisfied and all exit criteria must be completed for successful completion of this phase of the SI&T process.

8.2.1 Entry Criteria

Entry criteria for the Performance Test phase are the completion, baseline approval, and submission to CM of the project STP and STD and authorization from the Performance Test TRR to proceed.

8.2.2 Exit Criteria

Criteria for exit from the Performance Test phase and entry into the SQT phase are:

a. Review and approval of a Performance Test STR.

b. Submission of SPR(s) for all detected problems.

c. Authorization of the SQT TRR to proceed to the SQT phase.

d. System submitted to CM.

8.3 Test Group Activities

The Test group acts as the lead group and is responsible for system testing during the Performance Test phase.

All tests conducted during the Performance Test phase will be of a formal "black box" type. The Test group, following the STP and STD guidelines, executes all test case procedures in accordance with the associated test case and records all testing results, as they are observed.

Both online and batch processing parameters will be verified. The Test group will perform system configuration, environment, and disk space analysis required to verify the system production environment is implemented or simulated for this test phase.

The Test group will verify the configuration, size, and content of all databases used during the Performance Test phase, to verify they fulfill the SRD, STP, and/or SSS requirements. If databases do not conform to configuration, size, and content requirements, the Test group will work in cooperation with the SD group, prior to the beginning of this test phase, to create databases that fulfill all requirements.

All problems encountered during the Performance Test phase of the SI&T process will be documented and reported using an SPR form. After initialization of an SPR, it shall be submitted for review and analysis by the SPR review phase. Refer to U.S. Department of Education, SFA, document "Procedures For Using Testing Tool," Section 2.7, for an explanation of the SPR review process. For the template and instructions for completion of the SPR form, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix D.

If the SD group resolves a reported problem caused by the software, the test case procedure or test script that caused the problem will be re-tested using the original test case. If the repair to the software is tested and found not to satisfy the Pass/Fail criteria established by the STD for the function being repaired or tested, the Test group will not recommend the SPR for closure. Upon completion of the re-test, the SPR will be returned to the SPR Review process with the recommendation that the SPR be returned to the SD group for further investigation and repair. The process of repair and test will be repeated until the implemented repair allows the repaired function to satisfy the STD-established Pass/Fail criteria. When the implemented repair allows the repaired function to satisfy the STD-established Pass/Fail criteria, the Test group will recommend to the SPR Review process that the SPR status be changed to closed.

Upon completion of the Performance Test phase, an STR will be created by the Test group, with the cooperation of the SD and QA groups, and submitted to the SQT TRR and the CM group.

For the template and instructions for completion of the STR form, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix C.

If problems occur during the Performance Test phase activities that require a change(s) and/or modification(s) to a governing standard, plan, template, or guideline, CM must be notified prior to implementation of the required change(s) and/or modification(s). Test group personnel will provide CM with a description of the problem, the cause of the problem, and the documentation change(s) required. No change(s) and/or modification(s) to a governing standard, plan, template, guideline, or previously created documentation under CM version control shall be undertaken until notification of approval for the change(s) is received from the CM group.

Test group personnel will participate in the SQT TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of Test group activities during the Performance Test phase of the SI&T process, refer to U.S. Department of Education, SFA, documents "Procedures and Templates For Test Execution," Section 7, and "Procedures For Using Testing Tool," Section 2.

8.4 System Development Group Activities

The SD group will work in cooperation with the Test group, prior to the beginning of this test phase, to create databases, if necessary, that meet configuration, size, and content requirements of all databases. All databases used in the Performance Test phase will conform to requirements stated in the SRD, STP, and/or SSS.

During the Performance Test phase, the SD group will coordinate its activities with the Test group to provide support and be prepared to expedite the repair of all SPRs that require changes, modifications, and/or upgrades to the system.

If problems occur during the Integration Test phase activities that require a change(s) and/or modification(s) to a governing standard, plan, template, or guideline, CM must be notified prior to implementation of the required change(s) and/or modification(s). SD group or developer personnel will provide CM with a description of the problem, the cause of the problem, and the documentation change(s) required. No change(s) and/or modification(s) to a governing standard, plan, template, guideline, or previously created documentation under CM version control shall be undertaken until notification of approval for the change(s) is received from the CM group.

At the end of the Performance Test phase, the SD group will coordinate and cooperate with the Test group in the creation of the STR.

SD group personnel will participate in the SQT TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

8.5 Configuration Management Group Activities

The CM group will place the baseline system source code and HWCI under version control and assign version identification to the test artifacts, documents, and documentation, as received or at the completion of the Performance Test phase. All "closed" SPRs will be submitted to CM and will be classified as documents for CM accounting purposes.

CM will approve and track, by version control, all changes and/or modifications to all previously received governing standards, plans, templates, guidelines, documents, and documentation that occurred during the Performance Test phase. All version control changes and associated information will be reported to the PM in a timely manner and will become a part of the record of the SQT TRR.

CM group personnel will participate in the SQT TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of CM group activities during the Performance Test phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 4.

8.6 Quality Assurance Group Activities

The QA group will monitor the activities of the Test, CM, and SD groups or developers to verify compliance with all Performance Test phase standards, plans, templates, and guidelines. Formal or informal inspections or receipt of standard documentation will verify activity compliance.

The QA group will certify the proper system configuration and environment as required in the STP.

At the end of the Performance Test phase, the QA group will coordinate and cooperate with the Test group in the creation of the STR.

QA group personnel will participate in the SQT TRR that convenes at the end of this test phase. For a checklist template for actions required to complete this review, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix B.

For a detailed description of QA group activities during the Performance Test phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 7.

This page left intentionally left blank

9. SYSTEM QUALIFICATION TEST, PHASE FIVE

9.1 Phase Purpose

The SQT phase is the final phase of the SI&T process prior to customer/user acceptance of the software or system. The SQT verifies system functional performance in accordance with all relevant specifications and requirements used for the design and implementation of the software or system. Where all previous SI&T phases were of a more technical nature and performed by technical specialists who are close to the details of the system, the SQT phase focuses on the functionality of the software or system.

The STO will work with senior users of the proposed system to develop test cases and test case procedures appropriate for the SQT. The extent of user participation in the SQT phase of the SI&T process will be determined by the PM and the STO.

To verify functionality of the totally integrated software or system, testing during the SQT phase includes all functional areas and interfaces. If possible, experienced system software users will work with the STO to develop appropriate test cases and test case procedures. All software or system functions and subsystem interfaces will be independently tested in a systematic manner.

The results of the SQT will be documented and included in the FSTR. The FSTR will then be distributed for review and approval. The FSTR will serve as evidence that the tested software or system has met all requirements as stated in the SRD, STP, and STD and will be used as documentation of system acceptance by the PM.

Security requirements, if necessary, shall be considered during the performance of all activities during this phase.

9.2 Entry and Exit Criteria

Entry criteria must be satisfied and all exit criteria must be completed for successful completion of this phase of the SI&T process.

9.2.1 Entry Criteria

Entry criteria for the SQT phase shall be the completion, baseline approval, and submission to CM of the project STP and STD and authorization from the SQT TRR to proceed.

9.2.2 Exit Criteria

Criteria for exit from the SQT phase and completion of the SI&T process are the approval of the FSTR and acceptance by the PM and/or the customer of the tested system.

9.3 Test Group Activities

The Test group will act as the lead group and will be responsible for software or system testing during the SQT phase.

During the SQT phase, users will participate in developing and recording test cases, test case procedures, and data for system testing. Users will participate in the performance of test cases during the SQT phase.

All tests conducted during the SQT phase will be of a formal "black box" type. The users and Test group, following STP and STD guidelines, will execute all test case procedures in accordance with the associated test case and record all testing results, as they are observed.

SQT test cases will be adaptations of previously used test cases and test case procedures. SQT components and objectives will include the following areas:

- Functional Tests - Tests that cover functional SRD and STP requirements.
- Single User Tests - Tests performed for each functional area to validate the program operation in a one-on-one link.
- SPR Correction/Closure Testing - Tests executed to verify fixes to problems and verify the recommendation to close the SPR.

All problems encountered during the SQT phase of the SI&T process will be documented and reported using an SPR form. After initialization of an SPR, it shall be submitted for review and analysis by the SPR review phase. Refer to U.S. Department of Education, SFA, document "Procedures For Using Testing Tool," Section 2.7, for an explanation of the SPR review process. For the template and instructions for completion of the SPR form, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix D.

If the SD group resolves a reported problem that was caused by the software, the test case procedure or test script that caused the problem will be re-tested using the original test case. If the software repair is tested and found not to satisfy the Pass/Fail criteria established by the STD for the function being repaired or tested, the Test group will not recommend the SPR for closure. Upon completion of the re-test, the SPR will be returned to the SPR Review process with the recommendation that the SPR be returned to the SD group for further investigation and repair. The process of repair and test will be repeated until the implemented repair allows the repaired function to satisfy the STD-established Pass/Fail criteria. When the implemented repair allows

the repaired function to satisfy the STD-established Pass/Fail criteria, the Test group will recommend to the SPR Review process that the SPR status be changed to closed.

At the completion of the SQT phase, all test resources created from the use of automated test tools will be submitted to CM.

Upon completion of the SQT phase, the Test group, with the cooperation of the SD and QA groups will create an FSTR. The FSTR will be distributed for approval to the customer/user and the STO. When all required groups or organizations have approved the FSTR, it will be submitted to the CM group. For the template and instructions for completion of the FSTR form, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Test Execution," Appendix C.

If problems occur during the SQT phase activities that require a change(s) and/or modification(s) to a governing standard, plan, template, or guideline, CM must be notified prior to implementation of the required change(s) and/or modification(s). Test group personnel will provide CM with a description of the problem, the cause of the problem, and the documentation change(s) required. No change(s) and/or modification(s) to a governing standard, plan, template, guideline, or previously created documentation under CM version control shall be undertaken until notification of approval for the change(s) is received from the CM group.

For a detailed description of Test group activities during the SQT phase of the SI&T process, refer to U.S. Department of Education, SFA, documents "Procedures and Templates For Test Execution," Section 9, and "Procedures For Using Testing Tool," Section 2.

9.4 System Development Group Activities

During the SQT phase, the SD group will coordinate its activities with the Test group to provide support and be prepared to expedite the repair of all SPRs that require change, modifications, and/or upgrades to the software or system.

If during the SQT phase, activities, problems occur that require a change(s) and/or modification(s) to a governing standard, plan, template, or guideline, CM must be notified prior to implementation of the required change(s) and/or modification(s). SD group or developer personnel will provide CM with a description of the problem, the cause of the problem, and the documentation change(s) required. No change(s) and/or modification(s) to a governing standard, plan, template, guideline, or previously created documentation under CM version control shall be undertaken until notification of approval for the change(s) is received from the CM group.

At the end of the SQT phase, the SD group will coordinate and cooperate with the Test group in the creation of the FSTR.

9.5 Configuration Management Group Activities

The CM group will place the baseline system source code and HWCI under version control and assign version identification to the test artifacts, documents, and documentation, as received or at the completion of the Performance Test phase. All "closed" SPRs will be submitted to CM and will be classified as documents for CM accounting purposes.

CM will approve and track, by version control, all changes and/or modifications to all previously received governing standards, plans, templates, guidelines, documents, and documentation that occurred during the Performance Test phase. All version control changes and associated information will be reported to the PM at the end of the SQT phase.

At the completion of the SQT phase, all test resource files created by automated test tools will be submitted to the CM group.

At the end of the SQT phase, the CM group will coordinate and cooperate with the Test group in the creation of the FSTR.

The CM group will provide a production-ready version of the software or system and forward production ready reports and documents, including the FSTR, to the PM. All documentation from the SI&T process will be transferred in accordance with CMO standards and procedures to the CMO for proper archiving and storage.

For a detailed description of CM group activities during the SQT phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 4.

9.6 Quality Assurance Group Activities

The QA group monitors activities of the Test, CM, and SD groups or developers to verify compliance with all Performance Test phase standards, plans, templates, and guidelines. Formal or informal inspections or receipt of standard documentation will verify activity compliance.

The QA group certifies the proper system configuration and environment, as required in the STP.

At the end of the SQT phase, the QA group will coordinate and cooperate with the Test group in the creation of the FSTR.

The QA group will verify the FSTR was properly reviewed and approved before submission to the CM group.

For a detailed description of QA group activities during the SQT phase of the SI&T process, refer to the U.S. Department of Education, SFA, document "Procedures and Templates For Configuration Management and Quality Assurance," Section 7.

This page left intentionally left blank

GLOSSARY

Aggregate

A mass of distinct things gathered into a total or whole.

Aggregation Level

Effective measurement analysis and reporting requires that the data be aggregated to higher levels of the software components and project organizational structure. The aggregation levels define the different ways the measurement data can be grouped and organized for reporting on the project. The aggregation levels describe how the measurement data relates to an existing product and process structures. The organization that allows the measurement results to be combined, and later decomposed, into meaningful pieces of information.

Aggregation Structure

The structure used to define the data according to the defined aggregation levels. The levels may describe the personnel and management structure of the project, or the configuration of physical components of the project. All entries in a structure should be of the same type, such as software modules. However, these entries may reside at various levels of the structure, such as software modules at the unit level, CSCI, or integrated level of the software architecture.

Application

(1.) A complete, self-contained program that performs specific function(s) directly for the user.

(2.) In the TPM process this term refers to one of the two basic measurement activities, which comprise the system measurement process. The application activity involves collecting, analyzing, and acting upon the measurement data.
See **Tailoring**

Automated Test Script

A computer readable set of instructions that perform a sequence of steps, sub-steps, or other actions, performed serially, in parallel, or in some combination of consecution, that creates the desired test conditions that the test case is deigned to evaluate.

Baseline

A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

Baseline Control

Baseline control is the process that regulates approved and released versions of all software, documentation, and the test environment throughout the test life cycle.

Black Box Testing

This is testing associated with functional testing where the object being tested is treated as a black box. In this type of testing the test object is subjected to inputs and outputs that are verified for conformance to prescribed specifications.

Capacity Testing

Attempts to simulate expected customer peak load operations in order to ensure that the system performance requirements are met. It does not necessarily exercise all of the functional areas of the system, but selects a subset that is easy to replicate in volume. It will ensure that functions, which are expected to use the most system resources, are adequately represented.

Change Control

The process by which problems and changes to the software, documentation, and test environment are evaluated, approved, rejected, scheduled, and tracked.

Computer Aided Software Engineering (CASE)

A technique for using computers to help with one or more phases of the software life cycle, including the systematic analysis, design, implementation and maintenance of software. Adopting the CASE approach to building and maintaining systems involves software tools and training for the developers who will use them.

Computer Software Configuration Item (CSCI)

An aggregation of software that is designated for CM and treated as a single entity in the CM process.

Configuration Control

An element of CM, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

Configuration Item (CI)

Hardware or software, or an aggregate of both, which is designated by the project configuration manager (or contracting agency) for CM.

Configuration Management (CM)

A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements.

Configuration Management Office (CMO)

The Configuration Management Office (CMO) is the officiator of the project life cycle CM process.

Criteria

A standard, rules, or tests by which something can be judged.

Critical Defect

See **Criticality**

Criticality

The assessment of the impact upon a system of a given error, defect, problem, or discrepancy during the life cycle of a system.

The definition of critical and non-critical system defects or problems should be addressed at a management level and can be different for each system. For any given system error, defect, problem, or discrepancy, an appropriate impact value (i.e., priority) will be assigned.

An example of impact values with the corresponding priority numbers is presented below, as contained in IEEE/EIA Std-12207, 1998. The priority that will apply if a problem can result in one or more of these impacts:

PRIORITY	IMPACT
1.	a.) Prevent the accomplishment of an operational or mission essential capability. b.) Jeopardize safety. c.) Cause significant technical, cost, or schedule risks to the project or to life cycle support of the system.
2.	a.) Adversely affect the accomplishment of an operational or mission essential capability and no work-around solution is known. b.) Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, and no work-around is known.
3.	a.) Adversely affect the accomplishment of an operational or mission essential capability, but a work-around solution is known. b.) Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, but a work-around is known.
4.	a.) Results in user/operator inconvenience or annoyance, but does not affect a required operational or mission essential capability. b.) Results in inconvenience or annoyance for development or support personnel, but does not prevent the accomplishment of the responsibilities of these personnel.
5.	a.) This priority denotes any other effect.

Customer

The organization that procures software systems for itself or another organization.

Developer

An organization that develops software products. The term “develop” may include develop, modification, integration, reengineering, sustaining engineering, maintenance, or any other

activity that results in software products. The developer may be a contractor or a government agency.

Discrepancy

An inconsistency or disagreement found during testing between the actual and expected test results.

Document

A data medium and the data recorded on it that generally has permanence and can be read by a human operator or machine. Often used to describe human readable items only (e.g., technical documents, design documents, requirements documents, etc.).

Documentation

(1.) A collection of documents on a given subject.

(2.) The management of documents, that includes the actions of identifying, acquiring, processing, storing, and disseminating.

(3.) Any written or pictorial information describing, defining, specifying, reporting or certifying activities, requirements, procedures, or results.

Driver

A software program that exercises a system or system component by simulating the activity of a higher-level component.

Emulation

One system is said to emulate another when it performs in exactly the same way, though perhaps not at the same speed. A typical example would be the emulation of one computer by (a program running on) another. You might use emulation, as a replacement for a system whereas you would use a simulation if you just wanted to analyze it and make predications about it.

Emulator

Hardware or software that performs emulation.

Entry Criteria

A set of decision making guidelines used to determine whether a system under test is ready to move into, or enter, a particular phase of testing. Entry criteria tend to become more rigorous as the test phases progress.

Environment

The infrastructure in which a system is executing, consisting of hardware, operating system software, interfaces, etc.

Exit criteria

A set of decision-making guidelines used to determine whether a system under test is ready to exit a particular phase of testing. When exit criteria are met, either the system under test moves on to the next test phase or the test project is considered complete. Exit criteria tend to become more rigorous as the test phases progress.

Final System Test Report (FSTR)

Used to determine whether system testing is completed and to assure that software is ready for production.

Hardware Configuration Item (HWCI)

An aggregation of hardware that is designated for CM and treated as a single entity in the CM process.

Independent Verification and Validation (IV&V)

The verification and validation of a software product by an organization that is both technically and managerially separate from the organization responsible for developing the product.

Indicator

A measure or combination of measures that provides insight into a system issue or concept. TPM frequently uses indicators that are comparisons, such as planned versus actual measures. Indicators are generally presented as graphs or tables.

Integration

Combining software or hardware components or both into an overall system.

Integration Testing

The period of time in the software lifecycle during which the application is tested in a simulated production environment to validate the communications and technical architecture of the system. This test phase occurs when all the constituent components of the system under test are being integrated.

Interactive Development Environment (IDE)

A system for supporting the process of writing software. Such a system may include a syntax-directed editor, graphical tools for program entry, and integrated support for compiling and running the program and relating compilation errors back to the source code.

Interface

(1.) A shared boundary (e.g., a hardware component linking two devices or registers, or a portion of storage accessed and/or modified by two or more computer programs).

(2.) To interact or communicate with another system component.

Interface Requirement

A requirement that specifies a hardware, software, or database element with which a system or system component must interface, or that sets forth constraints caused by such an interface.

Interface Specification

A specification that sets forth the interface requirements for a system or system component (e.g., the software interface specification document).

Interface Testing

Tests conducted to ensure that program or system components correctly pass data and/or control to one another.

Issue

An area of concern where obstacles to achieving program objectives might arise. Issues include risks, problems, and lack of information. These three types of issues are defined as:

- Risk -- An area of concern that could occur, but is not certain. A risk is a potential problem. Risks represent the potential for the realization of unwanted, negative consequences from a project event. For example, a project plan may be based on the assumption that a COTS component will be available on a given date. There is a possibility (probability) that the COTS may be delayed and have some amount of negative impact on the project.

- Problem -- An area of concern that a project is currently experiencing or is relatively certain to experience. For example, a shortage of staff with the right skills may be an actual problem that is delaying the project.

- Lack of Information -- An area where the available information is inadequate to reliably predict project impact. Thus, satisfaction of project objectives is questionable even if no problems or risks are present. For example, lack of information about the size of the software to be developed could result in the project "discovering" that it has more work to do than originally planned.

Measure

The result of counting or otherwise quantifying characteristics of a process or product. Measures are numerical values assigned to system attributes according to defined criteria.

Measured (or actual) Value

Actual, current measurement data, such as hours of effort expended or line of code produced.

Measurement

The process of assigning quantitative values of system properties, according to some defined criteria. This process can be based on estimation or direct measurement. Estimation defines planned or expected measures. Direct measurement results in actual measures.

Measurement Analysis

The uses of measurement data to identify problems, assess problem impact, project an outcome, or evaluate alternatives related to system issues.

Measurement Analyst

The person(s) or team responsible for tailoring and applying system measures for a given project or task.

Measurement Information

Knowledge derived from analysis of measurement data and measurement indicators.

Milestone

A scheduled event for which some project or task member or manager is held accountable. A milestone is often used to measure progress.

Module

A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.

Note: *The terms 'module', 'component', and 'unit' are often used interchangeably or defined to be sub-elements of one another in different ways depending on the context.*

Non-Critical Defect

See **Criticality**

Performance Testing

The period of time in the system or software development lifecycle during which the response times for the application are validated to be acceptable. The tests ensure that the system environment will support production volumes, both batch and on-line.

Priority

A measure of the elements of importance related to the repair of a system problem that are not considered in defining the severity of a system problem.

Project Manager (PM)

The official responsible for acquiring, developing, or supporting a system to meet technical, cost, schedule, and quality requirements. Acquisition, development, and support will include both internal tasks and work that is contracted to another source.

Quality Assurance (QA)

A planned and systematic pattern of all actions necessary to provide adequate confidence that the product optimally fulfils customer expectations.

Quality Control (QC)

The assessment of product compliance. Independently finding deficiencies assures compliance of the product with stated requirements.

Requirement

(1.) A condition or capability needed to solve a problem or achieve an objective.

(2.) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. The set of all requirements forms the basis of development.

Regression testing

Part of the test phase of software development where, as new modules are integrated into the system and the added functionality is tested, previously tested functionality is re-tested to assure that no new module has corrupted the system.

Risk

An area of concern that may occur, but is not certain. A risk is a potential problem. Risks represent the potential for the realization of unwanted, negative consequences from a project event. For example, a project plan may be based on the assumption that a commercial off the shelf (COTS) component will be available on a given date. There is a possibility (probability) that the COTS may be delayed and have some amount of negative impact on the project.

Severity

The degree to which a problem adversely influences the system's operation or the overall test effort.

Simulation

Attempting to predict aspects of the behavior of a system by creating an approximate (mathematical) model of it. This can be done by physical modeling, by writing a special-purpose

computer program or using a more general simulation package, aimed at a particular kind of simulation. Typical examples are aircraft simulators or electronic circuit simulators.

Simulator

Hardware or software that performs simulation.

Software Design Specification (SDS)

A document that records the design of a system or system component; typical contents include: system and/or component algorithms, control logic, data structures, data set use, input/output formats, and interface descriptions.

Software Development File (SDF)

The developer shall document the development of each Computer Software Unit (CSU), Computer Software Component (CSC), and CSCI in Software Development Files (SDF). The developer shall establish a separate SDF for each CSU or a logically related group of CSUs, for each CSC or a logically related group of CSCs, and for each CSCI. The developer shall document and implement procedures to establish and maintain SDFs. SDFs may be generated, maintained, and controlled by automated means. To reduce duplication, SDFs should not contain information provided in other documents or SDFs. The set of SDFs shall include (directly or by reference) the following information:

- Design considerations and constraints.
- Design documentation and data.
- Scheduling and status information.
- Test requirements and responsibilities.
- Test case, test case procedures, and results.

Software Life Cycle

The phases a software product goes through between when it is conceived and when it is no longer available for use. The software life cycle typically includes the following: requirements, analysis, design, construction, testing (validation), installation, operation, maintenance, and retirement. The development process tends to run iteratively through these phases rather than linearly; several models (spirals, waterfall, etc.) have been proposed to describe this process. Other processes associated with a software product are: QA, marketing, sales, and support.

Software Management Plan

A project plan for the development of the software component of a system or for the development of a software product.

Software Requirements Document (SRD)

This is a formal document derived from the Software Requirements Specification (SRS) that sets forth the requirements, specifications, and standards for a system (e.g., a software product). Typically included are functional specifications and requirements, performance specifications and requirements, interface specifications and requirements, design specifications and requirements, and development requirements and standards.

Software Requirements Specification (SRS)

A specification that sets forth the requirements for a system component; (e.g., a software product). Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards.

Software Tool

Computer programs used to help develop, test, analyze, or maintain another computer program or its documentation.

Specification

Documentation containing a precise, detailed, verifiable description of particulars with respect to the requirements, design, function, behavior, construction, or other characteristics of a system or system component.

Stub

(1.) A dummy procedure used when linking a program with a run-time library. The stub routine need not contain any code and is only present to prevent "undefined label" errors at link time.

(2.) A local procedure in a remote procedure call (RPC). The client calls the stub to perform some task and need not necessarily be aware that RPC is involved. The stub transmits parameters over the network to the server and returns the results to the client/caller.

System

(1.) Any large program.

(2.) The entire computer system, including the input/output devices, supervisor program or operating system and possibly other software.

System Problem Report (SPR)

A form that is used to record a discrepancy discovered during the Integration Test, Performance Test and System Qualification Test phases of the SI&T process concerning a Computer Software Configuration Item, a software system or subsystem, other software related items, and associated documentation.

System Problem Report (SPR) Status Report

The System Problem Report Status Report is used during the SPR Status Review to determine if the SPRs are being processed appropriately and expeditiously.

System Testing

The period of time in the software lifecycle during which the implementation of each requirement is validated.

Tailoring

In the TPM process, this term refers to one of the two basic measurement activities, which comprise the system measurement process. The tailoring activity includes identification and prioritization of program issues, selection and specification of appropriate system measures, and integration of the measurement requirements to the developer's system process.
See **Application**.

Test

The process of exercising a product to identify differences between expected and actual behavior.

Test Artifacts

An item created during the system integration and test process that is preserved upon completion of the test process (e.g., test plans, requirements documentation, automated test scripts, and test documentation).

Test Case

A description of a test to be executed for or focused on a specific test aim.

Test Case Procedures

A sequence of steps, sub-steps, and other actions, performed serially, in parallel, or in some combination of consecution, that creates the desired test conditions that the test case is designed to evaluate.

Test Case (Setup) Suite

The steps required to configure the test environment for execution of a test case.

Testing Condition

System state or circumstance created by proceeding through some combination of steps, sub-steps, or actions in a test case.

Testing Environment

The infrastructure in which the test is performed, consisting of hardware, system software, test tools, and procedures.

Test Plan

In a test plan the general structure and the strategic choices with respect to the test to be executed are formulated. The test plan forms the scope of reference during execution of the test and also serves as an instrument to communicate with the customer of the test. The test plan is a description of the test project, including a description of the activities and planning, therefore it is *not* a description of the tests themselves.

Test Readiness Review (TRR)

Review conducted to determine whether a software test phase has been completed and to assure that the software is prepared for the next step in the formal integration and testing procedures. Software test procedures and results are evaluated, for compliance with the software testing requirements and system descriptions, for adequacy in accomplishing testing goals. Also, provides the forum for updating and revising operational and supporting documentation.

Test Resources

Aids that are used by a test tool for collecting, tracking and controlling information. This information is:

- Software requirements defined in the Software Requirements Document.
- Test requirements defined in the System Test Description.
- Automated test case scripts, as defined in the System Test Description.
- SPRs, as determined at each phase of the System Integration and Testing process.

This information is controlled by CM at the end of the SI&T process for use whenever further testing may be conducted, using a testing tool, during the remaining lifecycle of the software or system.

Test Tools

The software, hardware, systems, or other instruments that are used to measure and test an item.

Traceability

Degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor, successor, or master-subordinate relationship to one another (e.g., the degree to which the requirements and design of a given software component match).

Unit

The lowest element of a software hierarchy that contains one or more of the following characteristics:

(1) A unit comprising one or more logical functional entities.

(2) An element specified in the design of a computer software component that is separately testable.

(3) The lowest level to which software requirements can be traced.

(4) The design and coding of any unit can be accomplished by a single individual within the assigned schedule.

Unit Test

The process of ensuring that the unit executes, as intended. This usually involves testing all statements and branch possibilities.

Version

One of a sequence of copies of a system, each incorporates new modifications.

Version Identifier

A unique identifier assigned to baseline software, documentation, and test environment.

Version Control

The process by which all changes to the software, documentation, and test environment are compiled and built into a new version of the system.

Version Control Report

A report that details all changes and enhancements made to current version of the software, documentation, and test environment.

White Box Testing

This type of testing is associated with structural testing in which the testing can be characterized as being tied to implementation details, such as control methods, database design, coding details, and logic paths. The process of how an individual input is treated to produce a given output is ascertained. Structural testing is sometimes referred to as "clear box testing" since white boxes are considered opaque and do not really permit visibility into the code.

Work Breakdown Structure (WBS)

A work breakdown structure for software defines the software-related elements associated with program work, work activities, and products. Many measures are aggregated and analyzed at various WBS levels.

BIBLIOGRAPHY

Black, Rex. *Managing The Testing Process,* Redman, WA: Microsoft Press, 1999

Koomen, Tim and Pol, Martin. *Test Process Improvement*, Essex, England, UK: Pearson Education Limited, 1999

Evans, Michael and Marciniak, John. *Software Quality Assurance and Management,* New York, NY: John Wiley & Sons, 1987

Carnegie Mellon University, Software Engineering Institute. *The Capability Maturity Model: Guidelines for Improving the Software Process*, 1995.

Institute of Electrical and Electronics Engineers (IEEE). "Glossary of Software Engineering Terminology," IEEE-Std-610.12, 1990.

Institute of Electrical and Electronics Engineers (IEEE)/Electronic Industries Alliance (EIA). "Software Life Cycle Processes," IEEE/EIA Std-12207, 1998.

U.S. Department of Education. "SFA System Integration & Testing Approach, SFA Modernization," Undated.

U.S. Department of Education. "SFA Enterprise Configuration Management Approach, SFA Modernization," Undated.

U.S. Department of Education, SFA. "Test Performance Measurements," December 2000.

U.S. Department of Education, SFA. "Procedures and Templates for Test Creation," November 2000.

U.S. Department of Education, SFA. "Procedures and Templates for Test Execution," November 2000.

U.S. Department of Education, SFA. "Procedures and Templates for Configuration Management and Quality Assurance," December 2000

U.S. Department of Education, SFA. 'Procedures for Testing Tool," November 2000

Federal Systems Integration and Management Center (FEDSIM). "FEDSIM Writers Guide, Version 2," May 1994

Practical Software Measurement: A Foundation For Objective Project Management, Version 3.1, 17 April 1998, Office of the Under Secretary of Defense For Acquisition and Technology, Joint Logistics Commanders Joint Group On Systems Engineering

Free On-Line Dictionary Of Computing web site at www.foldoc.org

APPENDIX A

TESTING TECHNIQUES

This page left intentionally left blank

Testing Techniques

Software managers and test managers must be familiar with the various techniques employed in system testing. Application of testing techniques and the degree of testing varies from project to project. Software test planning should identify the following items:

- The specific techniques to be employed.
- When the techniques will be used in the testing process.
- To what degree the tests should validate overall system requirements.

Testing should be differentiated from the notion of debugging. The characterizations of each are presented in Table A-1.

TESTING	DEBUGGING
Uses known condition, predefined test case procedures, and has a predictable outcome. The only unpredictable outcome is whether or not a program passes the test.	Starts from an unknown initial condition and does not have a predictable outcome.
Tests are designed and scheduled. They are predictable, constrained, and formal by nature.	Intuitive and experimental and requiring detailed design knowledge, and freedom to perform deductive analysis.
A demonstration of correctness or error.	A programmer's vindication.

Table A-1 Testing Versus Debugging

Understanding that testing is a formal discipline, one can approach the task from the two ends of the testing spectrum. At one end of the spectrum is the concept of structural testing. Structural testing can be characterized as being tied to implementation details, such as control methods, database design, coding details, and logic paths. This form of testing is often referred to as "White Box" testing. SU testing tends to be structural in nature.

At the other end of the spectrum is the concept associated with functional testing where the object under test is treated as a "Black Box." In this strategy, the test object is subjected to inputs and resulting outputs are verified for conformance to a prescribed specification. All other testing conducted during the SI&T process tends to be functional in nature.

In developing a strategy for testing at each level of SD, a combination of structural and functional tests is recommended. Table A-2 recommends the application of the structural and functional techniques through SD.

Table A-2 Testing Phase Techniques			
Test Phase	**Test Level**	**Technique**	**Purpose**
Software Unit Test Phase	Unit Test and Unit Integration Test	Structural Path Testing	Static testing of logic branches to validate control structures, loop boundaries, and error recovery processing.
		Structural Boundary Condition Testing	Static testing of input and output parameter tolerances and accuracy of algorithm implementations.
		Functional User Input Syntax Validation	Verifies processing of user input data to ensure proper conversation to internal form and validate error message generation in response to invalid inputs.
	CSCI Qualification Test	Functional Test Bed Based On Equivalence Partitioning	Equivalence partitioning defines the minimum number of functional tests required to exercise the maximum number of intra-CSCI logic paths and data interfaces. Verifies CSCI functional capability against the SRD and STP.
		Functional User Input Syntax Validation	Valid and invalid inputs to uncover errors in user interfaces. Verify error-handling facilities, as stated in the SRD.
		Structural Boundary Condition Testing	Testing the intra-CSCI and peripheral interfaces to find errors in input and output data tolerances and verify CSCI data processing and limits are implemented correctly.

Table A-2 Testing Phase Techniques			
Integration Test Phase	Integration Test	Functional Test Bed Based On Equivalence Partitioning	Test inter-CSCI logic paths and data interfaces. Verifies system software functional capability against the SRD. Testing based on defined test case(s) and test case procedure(s) documented in the STD.
		Functional User Input Syntax Validation	Valid and invalid inputs to uncover errors in user interfaces. Verify error-handling facilities, as stated in the SRD. Testing based on defined test case(s) and test case procedure(s) documented in the STD.
		Structural Boundary Condition testing.	Testing of inter-CSCI and peripheral interfaces to find errors in input and output data tolerances and verify that the systems data processing and limits are correctly implemented. Testing based on defined test case(s) and test case procedure(s) documented in the STD.
	Regression Test	Regression Test Configuration and Environment.	Functional and/or structural tests developed specifically to test reported high-priority problems and/or critical algorithms. Used to verify the integrity of the production program after changes are made to the software. Testing based on defined test case(s) and test case procedure(s) documented in the STD.

Table A-2 Testing Phase Techniques			
Performance Test Phase	Performance Test	Functional Test Configuration and Environment	High-volume functional test bed execution to determine system software load capacity and ability to meet overall requirements stated in the SRD. Testing based on defined test case(s) and test case procedure(s) documented in the STD.
System Qualification Test Phase	System Qualification Test	Functional Test Configuration and Environment.	Functional tests developed to test the system in a production environment, as a customer/user would utilize the system. Testing based on defined test case(s) and test case procedure(s) documented in the STD.

The basic techniques are listed below:

- Test Path Coverage Analysis is a structural technique easily applied during the SU test phase. Path testing involves exercising selected logic paths through the control structure of the software. Algorithms that determine path selection and the generated outputs of a path are also validated using this methodology. Anything less than complete coverage means that untested code is being integrated into the system.
- Equivalence Partitioning is a strategy employed in developing functional tests. Developing a set of test cases that incorporates all possible combinations of inputs to the system is typically not possible due to time, cost, and resource constraints. Equivalence partitioning involves developing tests that incorporate a subset of the inputs that exercise the maximum number of paths and algorithms in the system software. For example, a test does not need to be generated for every way to cause a specific result, but you must generate at least one test that will exercise that system to cause that specific result.
- Boundary Condition Analysis employs testing boundary conditions in either a structural or functional format. This form of testing involves identifying and using input values that exercise the maximum and minimum boundaries by input of test values just above and below the specified range end points. The intent is to validate system input/output tolerances and associated algorithms. Tests can be developed based on the user perspective (functional) or from a simulation of peripheral inputs (structural).
- Input Syntax Validation is a functional technique used to validate the man-machine interface to the system. This form of testing involves exercising the man-machine

interface to ensure acceptance of valid inputs to the system and the response to invalid inputs as to direct and/or indirect side effects.

Note in Table A-2 that the underlying strategy moves from a predominantly structural form to a more functional form, as the system moves through the phases of the SI&T process. The table is not all inclusive and the reader is invited to become educated on detailed issues of software testing prior to developing a test plan, test cases, or test case procedures or performing software testing.

This page left intentionally left blank